WO0046754

Publication Title:

Method and apparatus for using a general three-dimensional (3D) graphics pipeline for cost effective digital image and video editing, transformation, and representation

Abstract:

A method in accordance with our invention permits the manipulation of a two-dimensional pixel array using a standard 3D graphics pipeline. The method comprises the steps of forming a wire mesh of the two-dimensional image. This wire mesh comprises a set of triangles, in which all of the vertices and sides of the triangles are located in the same plane in x, y, z coordinate space. The 2D image is then mapped into the mesh. The resulting object representation can then be manipulated in the same manner that 3D representations are manipulated by a 3D graphics pipeline

----

PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(54) Title: METHOD AND APPARATUS FOR COST EFFECTIVE DIGITAL IMAGE AND VIDEO EDITING, USING GENERAL 3D GRAPHICS PIPELINE
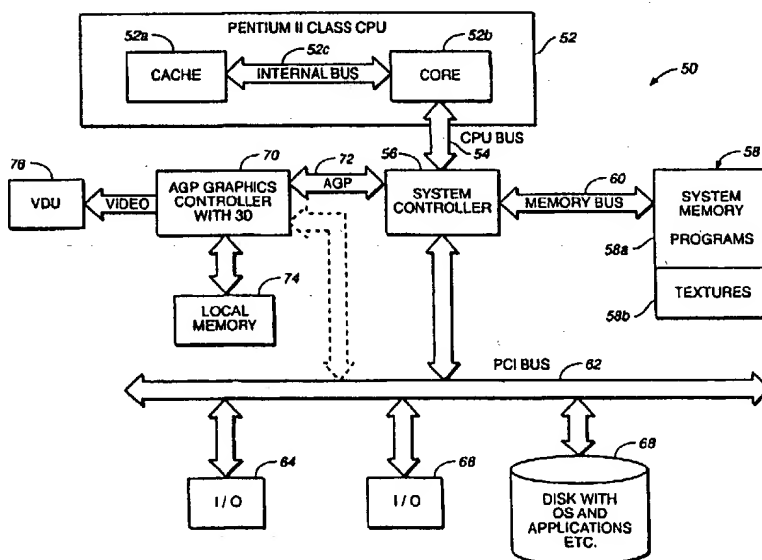
(57) Abstract

A method in accordance with invention permits the manipulation of a two-dimensional pixel array using a standard 3D graphics pipeline (70). The method comprises the steps of forming a wire mesh of the two-dimensional image. This wire mesh comprises a set of triangles, in which all the vertices and sides of the triangles are located in the same plane in x, y, z coordinate space. The 2D image is then mapped into the mesh. The resulting object representation can then be manipulated in the same manner that 3D representations are manipulated by a 3D graphics pipeline.

1 METHOD AND APPARATUS FOR COST EFFECTIVE DIGITAL IMAGE AND VIDEO EDITING, USING GENERAL 3D
2 GRAPHICS PIPELINE

3

4

5

6

7

8

9

10 BACKGROUND OF THE INVENTION

11   This invention pertains to digital image and video manipulation.

12   During the process of video and image editing, one alters portions or all of a

13 video image, e.g. by moving an image from one portion of a screen to another,

14 rotating the image, expanding the image or shrinking the image. Other common

15 image manipulations include warping an image (i.e. deforming the image in three

16 dimensions), magnifying parts of an image, performing cylindrical, spherical,

17 hyperbolic, or parabolic transformations to the image (e.g. causing a two-dimensional

18 image to appear to bend or wrap around a cylindrical, spherical, hyperbolic or

19 parabolic surface), melting images, etc.

20   Frequently, the image to be manipulated is in the form of a pixel array. In

21 other words, the image is stored as an array of data values, each value corresponding

22 to the color and brightness of a small area of the image. When performing the above-

23 mentioned video image manipulations, one typically performs calculations to

24 transform every pixel of the image. For example, for each pixel of the image, one

25 performs calculations to determine a new location for that pixel on a video screen.

26 Large images can contain millions of pixels. Further, for the case of image streams

27 (e.g. as in the case of video images), dozens of frames per second must be

28 transformed. Thus, many millions of pixels may have to be analyzed and modified

1   for each transformation in a very short amount of time. Such transformations require

2   either enormous CPU resources or special hardware to be able to transform large

3   numbers of pixels in a short amount of time.

4   It would be desirable to provide a method for editing and/or manipulating two-

5   dimensional images. This is particularly desirable so that one can edit and/or

6   manipulate many frames of two-dimensional images per second, e.g. a stream of

7   images such as video images. It would be desirable to be able to perform these

8   manipulations quickly and efficiently, without having to construct expensive,

9   sophisticated hardware dedicated to this purpose.

10

11   SUMMARY

12   A method in accordance with our invention comprises the step of transforming

13   2D images into 3D space and using a 3D graphics pipeline for manipulating the

14   image. This method frees up CPU resources, decreases the amount of time needed to

15   perform image manipulation, and permits new special effects.

16   In one embodiment of our invention, one obtains an image in the form of an

17   array of pixels. This can be accomplished, for example, by digitizing an image. One

18   then "transforms" these images into 3D space using a 3D graphics pipeline. The

19   graphics pipeline can be completely or partially implemented in software or in

20   hardware (e.g. a 3D hardware accelerator).

21   There are several ways of transforming the array of pixels into 3D space.

22   Typically, the 3D pipeline defines a set of image surface regions, each image surface

23   region being assigned a location in a three-dimensional coordinate system. However,

24   in the case of one embodiment of our invention, all of the surface regions are

2

1   coplanar. In one embodiment, the surface regions are all polygons, such as triangles,

2   each triangle being defined by a set of points that are assigned a location in terms of x,

3   y and z coordinates. (All of the coordinates are set up such that they are all located in

4   the same plane.) However, the surface regions can be defined using other graphics

5   techniques as well. For example, the surface regions or the whole surface can be

6   defined using an "implicit" technique, in which the positions of the pixels in the space

7   (surface regions) are defined by mathematical equations.

8   After the set of surface regions is created, we map the 2D pixel array image

9   onto the set of surface regions. In other words, we "bind" a portion of a picture of the

10 2D image to each of the surface regions. After performing this step, we use the 3D

11 graphics pipeline to manipulate the image we have created to perform desired image

12 and/or video editing.

13   We have discovered that this technique is extremely efficient for processing

14 many 2D images quickly. For example, this technique can be used to process a video

15 image stream, which typically comprises 59.94 frames per second, each frame

16 corresponding to a large pixel array. This technique can be used to manipulate 2D

17 images from other sources as well, e.g. digital scanners, cameras, image files

18 downloaded from the internet, or other sources.

19   In one embodiment, currently existing off-the-shelf 3D pipelines and/or 3D

20 enhanced processors are used to perform this process. Thus, we have discovered a

21 new, inexpensive method for manipulating 2D images without having to construct

22 special processing software and/or hardware from scratch.

23   It will be appreciated that we use a 3D graphics pipeline in a novel way.

24 Previously, graphics pipelines were used to map a texture onto a geometric object as a

3

1   method of giving the geometric object a particular appearance, and to create a new

2   virtual environment comprising a set of such objects. This was done by providing a

3   texture and a geometry to a graphics pipeline to create a new virtual image. This is in

4   contrast to using the geometry to transform and edit images.

5   During a method in accordance with our invention, we apply an image and a

6   geometry to a 3D graphics pipeline to generate a 3D image. We then use the 3D

7   graphics pipeline to transform or manipulate that image in a novel manner.

8   An image is not merely a cell in a pattern that is repeated over and over again,

9   as in some textures, nor is the image of a size such that it is barely perceptible. Rather

10 than simply applying a texture to a surface during the process of constructing an

11 image, we apply an image to a geometric surface to place that image in three-

12 dimensional space. The image can be a two-dimensional representation of a three-

13 dimensional object.

14

15 BRIEF DESCRIPTION OF THE DRAWINGS

16   Figs. 1A to 1E illustrate the operation of a 3D graphics pipeline.

17   Figs. 2A and 2B illustrate manipulation of a 2D image using a method in

18 accordance with our invention.

19   Fig. 3 is a simplified block diagram of a personal computer (PC) coupled to a

20 graphics controller with a 3D graphics pipeline in accordance with our invention.

21

22 DETAILED DESCRIPTION

23   As mentioned above, a method in accordance with our invention involves

24 using a 3D graphics pipeline in a novel manner to facilitate manipulation of digital

4

1   images, including video image streams. We will first explain how a 3D graphics

2   pipeline is normally used. Thereafter, we will describe its use during a method in

3   accordance with our invention.

4

5   <u>3D Graphics Pipelines</u>

6   The 3D graphics pipeline referred to in this patent can be implemented by a

7   combination of hardware elements, known as accelerators, and software, some of

8   which is sometimes referred to as drivers. The partitioning between hardware and

9   software may vary, depending upon the CPU used and the graphics card in the

10   system, but the overall system performs the method steps described below. Portions

11   of the pipeline tasks can be performed by software, which is less expensive than

12   hardware, but in general slower than hardware solutions at the present time. The

13   hardware and software that perform the steps described below are referred to simply

14   as a pipeline, without regard to the specific partitioning.

15   The following is a simplified, general description of 3D graphics pipelines. It

16   is not intended to describe any specific product (e.g. products mentioned later in this

17   patent). Rather, the following description is merely a general explanation of 3D

18   graphics pipelines to assist the reader's understanding.

19   Currently, graphics objects created using a 3D graphics pipeline can be

20   described as a set of geometric surfaces. One way of constructing a geometric surface

21   in a graphics pipeline is to create a "mesh" of "primitives." A "primitive" is a small

22   geometric surface that can be defined by a set of vertices. For example, the primitive

23   can be a polygon (e.g. a triangle or quadrilateral) defined within the pipeline in terms

24   of the locations (in x, y and z coordinate space) of its corners or vertices. A set of

25   several primitives is used to define a larger 3D surface.

5

1   Instead of using primitives, such as polygons, some graphics pipelines can

2   process geometric surface areas defined in other ways, e.g. by mathematical

3   equations. This technique for defining geometric surface areas is called "implicit."

4   As explained below, both techniques for defining such surface areas can be used to

5   perform a method in accordance with our invention.

6   For purposes of clarity of explanation, we will first describe a graphics

7   pipeline that processes geometric surface areas using triangular primitives. Other

8   types of graphics pipelines will be discussed later on.

9   In this first example, a 3D graphics pipeline constructs a 3D image of an

10   object from a 2D pixel array (typically called a "texture map"). Fig. 1A illustrates a

11   2D image 2 of a set of "textures." (As will be explained below, this texture map is

12   used to create the image of an object—in this case, a house. Image 2 includes a

13   portion 2a, which has the appearance of bricks, portion 2b, which has the appearance

14   of roof shingles, portion 2c, which has the appearance of a door, and portion 2d which

15   has the appearance of a window.) 2D image 2 is stored in a digital memory in the

16   form of an array of pixels. Each location in the memory stores a pixel, which is one

17   or more words of data indicating the color, color saturation and brightness

18   corresponding to that pixel. The location of the pixels within the array is typically

19   referred to as u, v coordinates (not to be confused with the Y, U and V signal names

20   used to describe certain video signals). (The u, v coordinates are similar to x, y

21   coordinates of the Cartesian coordinate system. In Fig. 1A, the pixel array is an n by

22   m array, where n and m are integers.)

23   As mentioned above, Fig. 1A represents a pixel array. Physically, the array

24   comprises data loaded into a memory.

SUBSTITUTE SHEET (RULE 26)

1   The next step in the process is to provide or prepare a geometric surface. In

2   this example, the geometric surface is in the form of a mesh 4 of primitives 5 in three

3   dimensional space (Fig.1B). In the case of Fig. 1B, the primitives are triangles, but

4   other types of polygons can be used. The mesh of primitives represents a three-

5   dimensional shape of an object O in 3D space (in the case of Fig. 1B, the shape of a

6   house). The position of each vertex of each triangle within mesh 4 is stored in a

7   memory in the form of x, y and z Cartesian coordinates, relative to the object. These

8   coordinates are sometimes referred to as model coordinates ("MC"). The process of

9   preparing such a mesh is well-known, and described in standard graphics libraries,

10   such as Real 3D, published by Real 3D, a Lockheed Martin Corporation, in 1996, and

11   Direct 3D, published by New Riders Publishing in 1997.

12   The mesh of Fig. 1B is not displayed as such. Rather, the mesh of Fig. 1B is a

13   representation of what is stored in a digital memory. Specifically, the memory stores

14   the locations, in terms of x, y and z coordinates, of each vertex within mesh 4.

15   The next step is to map or "bind" the two-dimensional texture map of Fig. 1A

16   onto mesh 4 of Fig. 1B. This is accomplished by mapping each triangle vertex to a

17   location in the texture map. In effect, a list of data points is prepared that associates

18   each vertex of mesh 4 to the u, v coordinates of a particular point (pixel) in the texture

19   map of Fig. 1A. (The locations in the texture map to which the vertices are bound are

20   sometimes referred to as "control points.")

21   This portion of the process is roughly analogous to an upholsterer choosing a

22   piece of fabric, and binding it with a few nails to the corner of a couch being

23   upholstered (the nails are like control points). The upholsterer subsequently asks his

7

1   apprentice to finish attaching the fabric to the couch. In this case, the 3D graphics

2   pipeline finishes the task instead of an apprentice.

3   Figs. 1A and 1B describe the process by which one texture map (Fig. 1A) is

4   mapped onto one mesh 4 representing one object O. A graphics pipeline can, and

5   often does, map one or several texture maps onto the same or several different objects.

6   The next step in the process is to set up a "world coordinate model" of the

7   various objects to be displayed. This requires establishing a position and directional

8   orientation for each object to be displayed. For example, supposing that there are to

9   be two objects to be viewed: a tetrahedron T and a cube C (Fig. 1C). During this step

10   of the process the pipeline is instructed that Cube C is to be facing in a certain

11   direction, and is to be located partially in front of tetrahedron T relative to a certain

12   frame of reference. Again, the structure of Fig. 1C is not displayed per se. Rather,

13   the graphics pipeline sets up processing of the model coordinates in accordance with

14   the parameters of the position and orientation of the object.

15   The next step is to select a frame of reference. For example, it might be

16   decided that the "viewer" will want to observe the objects from a position

17   corresponding to a corner of the world coordinate model (e.g. position P in Fig. 1D).

18   Thus, a virtual viewpoint, viewing direction and aperture will be selected. The

19   parameters associated with this "viewer" define the screen coordinate (SC) system.

20   Further, it might be decided the viewer will observe these objects with a light source

21   located at a position L. The graphics pipeline will set up another processing pipe to

22   process the world coordinate data into the screen coordinate data which will cause a

23   computer screen to display the image as it would be perceived by the observer at

24   position P (e.g. the image of Fig. 1D). In other words, the computer screen will

1   provide an image of tetrahedron T and cube C as they would be observed by a viewer

2   if he were standing at position P, and a light source were present at location L. This

3   image will be provided initially as a pixel array in a frame buffer and then displayed

4   by the computer screen. The image in the frame buffer is refreshed, i.e. regenerated

5   according to the specifications programmed into the pipeline, typically at about 50 to

6   120 times per second. There are many different methods for optimizing the pipeline,

7   and minimizing the time spent processing the invisible parts of the objects, such as the

8   backside of cube C facing away from the viewer. Such details are well-known to

9   those skilled in the art, and will not be discussed in detail here.

10   During the above-described process constructing the pixel array and providing

11   it in the frame buffer, the pipeline a) fetches the portion of texture map 2 "tacked" to

12   the vertices of mesh 4 (and therefore stretched over each triangle); b) determines how

13   and where that portion of the texture map should appear, given the orientation of the

14   triangles relative to the viewer and the location of the light source; and c) constructs

15   the appropriate bit map pixel array for storage in the frame buffer. The contents of

16   this frame buffer are then displayed as an image on a computer screen.

17   Thereafter, the 3D graphics accelerator permits one to manipulate the

18   displayed objects in any desired manner. For example, if one wants to rotate the

19   image of tetrahedron T by 45° (Fig. 1E), the 3D graphics accelerator facilitates this

20   manipulation. This is accomplished by providing a new set of parameters in the

21   world coordinate model for the graphics pipeline indicating the new position and

22   orientation for tetrahedron T. After this occurs, the next time the graphics pipeline

23   regenerates the image stored in the frame buffer, the regenerated image will reflect

24   this rotation of tetrahedron T.

1   Similarly, suppose that it is desired to display what would appear to the viewer

2   if he took ten steps forward from his location at position P. The next time the

3   graphics pipeline regenerates the image, it will generate and store another pixel array

4   in the frame buffer corresponding to what would appear to such a viewer, and this

5   pixel array is provided as another image on the computer screen.

6   It is thus seen that the graphics pipeline is extremely useful in applications

7   such as video games, where it is desired to simulate what would appear to a game

8   player if he were wandering past a set of objects.

9   As mentioned above, some graphics pipelines create models of geometric

10 surfaces using an implicit technique. These surfaces are often described as a function

11 of the position coordinates, i.e. $f(x,y,z)$, or can also contain some vertices. Control

12 points and additional formulas associated with such surfaces are used to bind a digital

13 pixel array (e.g. an array as shown in Fig. 1A) to the implicitly defined surface, and

14 the process proceeds as described above. The major difference is that instead of

15 defining surface areas in terms of primitives with vertices, the surface areas are

16 defined in terms of mathematical equations.

17

18 <u>Manipulation of 2D Images</u>

19   A process in accordance with our invention begins with the step of obtaining a

20 two-dimensional digital image (e.g. image 10 in Fig. 2A). This step can be

21 performed, e.g., by scanning an image such as a photograph or other picture using a

22 conventional digital scanner. The digital image can also be obtained from a

23 conventional digital camera. The image can also consist of digital video image, e.g.

24 out of a live or stored video stream, which is basically a fast succession of 2D images.

1   However, any other source of a 2D digital image can be used. As mentioned above,

2   the digital image is typically stored in a memory as an array of digital values. In one

3   embodiment, the digital values are in a compressed form, e.g. using a compression

4   technique such as MPEG1 or MPEG2 or other formats. In the case of compressed

5   digital values, they must first be decompressed prior to processing. Also, scanned

6   images or digitized images from any source such as cable TV, antennas, cameras, etc.

7   can be used.

8   As mentioned above, for the case of video images, dozens of frames per

9   second comprising millions of pixels per second must be processed. We discovered

10   that standard graphics pipelines can be used to process frames of data sufficiently fast

11   to process video images.

12   Any type of memory can be used to store the digital 2D image, e.g.

13   semiconductor memories (SRAMs, DRAMs or other semiconductor memories), a

14   magnetic memory (e.g. a hard disk, a floppy disk, magnetic tape, or magneto-optic

15   disk), or other type of memory device (e.g. an optical disk). The pixels corresponding

16   to the stored image can be stored in terms of RGB values (e.g. the strength of the red,

17   green and blue components of the pixel color), YUV values or other values. (For

18   YUV values, Y corresponds to the amplitude or brightness of the pixel value, U

19   corresponds to the color and V corresponds to the saturation.) The pixel values can be

20   encoded in other ways as well. Depending on the situation, a conversion may be

21   required before further processing.

22   Next, a 3D graphics pipeline is set up. This is accomplished by providing

23   instructions to the 3D graphics pipeline as to what is to be done with the data that is to

1   be provided. Setting up graphics pipelines per se is well known in the art, e.g. as

2   described in the Microsoft Direct 3D SDK (software developer kit) or Direct 3D.

3   Thereafter, a computer model of a planar geometric surface is generated. This

4   computer model can comprise a set of primitives, e.g. polygons such as triangles. In

5   another embodiment, the computer model can comprise an implicit description of a

6   flat geometric surface. This implicit description is typically a mathematical function

7   (e.g. a function of x, y and z) as described above.

8   For the case in which the planar geometric surface comprises a mesh of

9   primitives, the number and shape of primitives and the type of primitives can vary.

10   Fig. 2B illustrates a mesh 12 that can be used to practice a method in accordance with

11   our invention. Mesh 12 is similar to mesh 4 described above. However, unlike mesh

12   4, all of the vertices of mesh 12 are coplanar (or substantially coplanar). In one

13   embodiment, mesh 12 comprises about 5000 triangles, which would be acceptable for

14   processing a video image. Of course, other numbers of primitives could be used.

15   After constructing the planar geometric surface (e.g. mesh 12), image 10 is

16   mapped, or bound, onto the flat geometric surface. This is accomplished in the

17   following way. For the case in which the flat geometric surface is a mesh such as

18   mesh 12, each vertex of the flat geometric surface (e.g. the triangle vertices) is

19   associated with an image pixel location (i.e. control point). Thus, each control point

20   is associated with a texture coordinates (u, v) corresponding to a pixel. A table of

21   data listing each vertex and its associated u, v texture space coordinates is set up.

22   This is called "binding." (See Kamen, IEEE Computer Society, IEEE Computer

23   Graphics and Applications, Jan.-Feb. 1997, Vol. 17, No. 1.) For the case in which an

24   implicit technique is used to define the flat geometric surface, control points within

1   the implicitly defined surface are bound to pixel array coordinate space (u, v

2   coordinates) in a manner analogous to the triangles discussed above.

3   After image 10 is mapped into mesh 12, the object can be manipulated by

4   manipulating the world coordinates. The world coordinates describe where in the x,

5   y, z space the plane is to appear, and what its orientation will be (i.e. what angle it

6   should be held at with respect to the viewer). In addition, the screen coordinates for

7   the object can be changed. As a result, when the 2D image is finally prepared, it can

8   be prepared in such a manner that reflects the desired manipulation. For example, it

9   can be rotated about any axis, magnified, shrunk, etc.

10   After establishment of the world coordinate model and screen coordinate

11   model, the pipeline prepares an array of pixels in the output frame buffer (OFB),

12   including pixels showing the manipulated textured mesh 12. The array of pixels in

13   the OFB is displayed on a CRT or other type of screen.

14   One can manipulate the video image by, for example, changing the world

15   coordinate parameters, e.g. telling the pipeline to tilt the video image about any axis

16   (including an axis perpendicular to the screen or in the plane of the screen). Thus,

17   when the pipeline regenerates the pixel array in the OFB, the regenerated video image

18   will appear tilted about the selected axis. Since the pipeline will regenerate the image

19   at a preprogrammed rate according to the system used, live video will appear as live

20   video. That is because every time a new pixel array is generated, the texture map,

21   which contains the incoming video frame buffer, is reread and put through the

22   pipeline. Since the texture mapping process also contains features for pixel

23   interpolation, an automatic resolution adaptation occurs.

SUBSTITUTE SHEET (RULE 26)

1   One can bend or warp the image by moving the vertices about which the

2   image is mapped. Thus, one can alter the flat geometric plane of Fig. 2B to thereby

3   warp the image. When the pipeline regenerates the pixel array in the frame buffer, the

4   image will appear warped.

5   One can move the vertices so that mesh 10 becomes a cylinder. When the

6   pipeline regenerates the pixel array in the frame buffer, the image will appear

7   wrapped around a cylinder. (Of course, the mesh 10 can be altered into other shapes,

8   and the image would be wrapped around the other shape.) These modifications could

9   be done at a speed that would create the impression in the viewer that the image was

10   being wrapped or warped gradually.

11   One could magnify or shrink images by moving vertices away from or closer

12   to each other, or moving the image closer or further from the viewer in the world

13   coordinate system, or by re-parameterizing the model coordinate to world coordinate

14   conversion.

15   One could cause the image to "melt" by causing the vertices to drop at

16   different rates.

17   The number of 3D manipulations possible is only limited by the energy of a

18   user to try them all out.

19

20   <u>Hardware and Software for Practicing An Embodiment of a Method In Accordance</u>
21   <u>With Our Invention</u>
22

23   One embodiment of our invention can be practiced using a PC having the

24   following:

14

1   1.   A CPU such as a Celeron or Pentium, e.g. as manufactured by Intel, or a K6

2   processor, e.g. as manufactured by Advanced Micro Devices.

3   2.   32 MB of memory or greater.

4   3.   A 3D HW adapter. This is a type of graphics card currently available on the

5   market. The 3D HW adapter should have 4 MB of memory (preferably 8 MB)

6   and an advanced graphics port (AGP) interface. (An AGP interface is a type

7   of bus standard that is well-known in the art.) Alternatively, a peripheral

8   connection interface ("PCI") can be used in lieu of a AGP. The PCI is a type

9   of bus standard that is well known in the art.   Examples of appropriate 3D

10   HW adapters include the TNT-2 available from Riva, the ATI Rage 128, the

11   Matrox G400, the Trident Blade 3D and the S3 Savage.

12   4.   The operating system can be Windows 95, Windows 98, Win2000, or any

13   other operating system that supports direct 3D. The Windows operating

14   system includes a standardized platform called Direct X for Windows.

15   In one embodiment, a user sets up the flat geometric surface (for example, a

16   triangle mesh) in the Direct 3D windows environment. The set of instructions is then

17   provided to the graphics pipeline, which finishes the rendering process. However, in

18   another embodiment, the PC comprises a bypass mechanism that permits one to

19   access the hardware accelerator directly using a software interface provided by the

20   graphics card manufacturer.

21   Fig. 3 is a block diagram of a computer system 50 for performing a method in

22   accordance with our invention. Referring to Fig. 3, system 50 comprises a CPU 52,

23   e.g. a Pentium II class CPU, comprising a cache memory 52a, a core 52b and an

24   internal bus 52c for facilitating communication between core 52b and cache 52a.

1   Core 52b communicates via a CPU bus 54 to a system controller 56. System

2   controller 56 communicates with the system memory 58 via a memory bus 60.

3   System memory 58 includes a first portion 58a which stores system memory

4   programs and a second portion 58b that stores the texture maps such as described

5   above.

6   Also included in system 50 is a PCI bus 62 for facilitating communication

7   between system controller 56 and I/O devices 64, 66 and disk drive 68. I/O device 64

8   can be any type of I/O device. In one embodiment, I/O device 66 is a video capture

9   card with a driver. Data from the video capture card is either loaded by DMA (direct

10   memory access) or CPU 52 into a frame buffer, typically within main memory 58.

11   However, the frame buffer may be in other memories within system 50.

12   System 50 also includes an AGP graphics controller 70 comprising a 3D

13   accelerator. In one embodiment, AGP graphics controller 70 communicates with

14   system controller 56 via an AGP bus 72. In an alternative embodiment, AGP graphics

15   controller 70 can communicate with system controller 56 via PCI bus 62 (e.g. as

16   shown in phantom in Fig. 3).

17   Graphics controller 70 uses its own local memory 74 to generate and store

18   pixel arrays to be displayed on a video display unit 76.

19   It is emphasized that system 50 is only one example of a system that performs

20   a method in accordance with our invention. Other hardware can be used as well.

21

22   Applications for the Invention

23   There are many applications for our invention. For example, a method in

24   accordance with our invention can be used to manipulate image streams such as

16

1　television images. The method of our invention is particularly appropriate since video

2　images comprise a succession of frames at a rate of about 60 frames/second in North

3　America. For instance in the case of NTSC, about 9.1 Mbytes per second throughput

4　are required. (NTSC is an abbreviation of "North American Television Standard for

5　Color." It is the standard used for television signals in North America.)

6　　　　Another application for our invention is to move and tilt portions of different

7　video images or other images onto different portions of a screen such as a PC screen.

8　The images, for example, could be transformed to appear on the faces of a polyhedron

9　(e.g. a cube). By clicking on appropriate icons on a computer display, one could turn

10　the polyhedron to see the different images on the various faces of the polyhedron.

11　The polyhedron could be used as a new type of computer menu option display.

12　　　　The manipulated image provided in accordance with our invention could be

13　provided to any appropriate output device, e.g. a television screen, a video projector, a

14　HDTV monitor, or a PC screen. The image manipulated in accordance with our

15　invention could come from any of a number of sources, e.g. an analog or digital video

16　input, a cable TV input, a satellite input, the internet, a digital scanner, a digital

17　camera, or numerous other sources. (In the case of an analog input, one would first

18　digitize the image.)

19　　　　While the invention has been described with respect to a specific embodiment,

20　those skilled in the art will realize that changes can be made in form and detail

21　without departing from the spirit or scope of the invention. For example, the

22　invention can be used to map 2D images onto substantially planar surfaces, rather

23　than exactly planar surfaces. Accordingly, all such changes come within the

24　invention.

17

SUBSTITUTE SHEET (RULE 26)

1   We claim:

2   1.   A method comprising:

3   providing a digital representation of a two-dimensional image;

4   providing a model of a geometric surface;

5   binding said digital representation to said model of said geometric surface; and

6   generating an array of pixels corresponding to an image to be displayed, said

7   image to be displayed including said two-dimensional image applied to said

8   geometric surface.

9

10 2.   Method of claim 1 further comprising the act of displaying said image to be

11 displayed on a display device.

12

13 3.   Method of claim 1 wherein during said act of binding said geometric surface is

14 substantially flat.

15

16 4.   Method of claim 3 further comprising the act of modifying said geometric

17 surface so that it is no longer substantially flat prior to said act of generating said

18 array of pixels, whereby said displayed image does not appear substantially flat.

19

20 5.   Method of claim 2 further comprising the act of performing a digital

21 manipulation process, said digital manipulation process comprising one of warping,

22 resizing, melting, and rotating said image to be displayed.

23

18

SUBSTITUTE SHEET (RULE 26)

1   6.   Method of claim 2 further comprising the act of establishing the location and

2   position of said geometric surface in a world coordinate system, wherein said act of

3   generating said array of pixels takes into account the location and position of said

4   geometric surface in said world coordinate system.

5

6   7.   Method of claim 1 wherein said geometric surface comprises a set of

7   polygons.

8

9   8.   Method of claim 1 wherein said geometric surface is defined with an implicit

10 technique.

11

12 9.   Method of claim 1 wherein said act of generating is accomplished using a 3D

13 graphics pipeline.

14

15 10.   Method of claim 1 wherein said representation of said two-dimensional image

16 is obtained from a video image.

17

18 11.   Structure comprising:

19   a first memory for storing a digital representation of a two-dimensional image;

20   a second memory for storing a model of a geometric surface;

21   a third memory for storing a world coordinate system, said world coordinate

22 system representing a location and orientation of an object to be displayed; and

23   a graphics 3D pipeline for binding said model of said geometric surface to said

24 two-dimensional image and generating a pixel array corresponding to an image to be

1   displayed, said generating of said image being based, at least in part, on said

2   geometric surface textured in accordance with said two-dimensional image, said

3   image being oriented and positioned in accordance with said world coordinate system.

4

5   12.   Structure of claim 11 further comprising a display for displaying said image to

6   be displayed.

7

8   13.   Structure of claim 11 wherein said geometric surface is substantially flat.

9

10   14.   Structure of claim 13 wherein said graphics pipeline permits manipulation of

11   said substantially flat geometric surface so that said image to be displayed does not

12   represent said geometric surface as substantially flat.

13

14   15.   Structure of claim 11 wherein said first, second and third memories comprise

15   different sets of locations within the same memory device.

16

17   16.   A method comprising the acts of:

18   providing an image to a 3D graphics pipeline;

19   providing a geometric surface to said 3D graphics pipeline; and

20   generating a pixel array based on the binding of at least a portion of said

21   image to said geometric surface.

22

23   17.   Method of claim 16 wherein said image is a video image.

SUBSTITUTE SHEET (RULE 26)

1   18.   Method of claim 17 wherein said geometric surface is a flat geometric surface

2   when said geometric surface is bound to said video image.

3

4   19.   A method comprising the acts of:

5   providing a two-dimensional image to a 3D graphics pipeline;

6   providing a substantially flat geometric surface to said 3D graphics pipeline;

7   generating a pixel array based on the binding of at least a portion of said two-

8   dimensional image to said flat geometric surface.

9

10 20.   Method of claim 19 comprising the acts of binding said two-dimensional

11 image to said geometric surface and thereafter altering said geometric surface so that

12 said geometric surface is no longer substantially flat.

13

14 21.   Method of claim 1 or 19 wherein said two-dimensional image is a digitized

15 photograph.

16

17 22.   Method of claim 1 or 19 wherein said two-dimensional image is a frame from

18 a video clip.

19

20 23.   Method of claim 1 or 19 wherein said two-dimensional image is a two-

21 dimensional image of a three-dimensional object.

22

23 24.   Structure comprising:

24   a source of a video image; and

21

1   a 3D graphics pipeline receiving said video image and processing said video

2   image by binding said video image to a geometric surface and generating a pixel array

3   based on the binding of at least a portion of said video image to said geometric

4   surface.

5

6   26.   Structure comprising:

7   a memory containing a representation of a two-dimensional image; and

8   a memory containing a geometric surface; and

9   a 3D graphics pipeline for binding said two-dimensional image to said

10   geometric surface and generating a pixel array corresponding to said bound two-

11   dimensional image.

12

13   27.   Structure of claim 26 wherein said geometric surface is substantially flat.
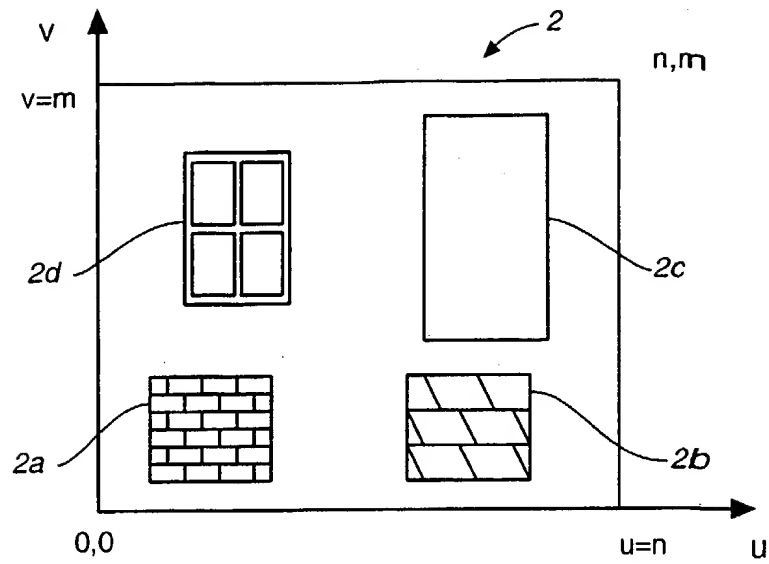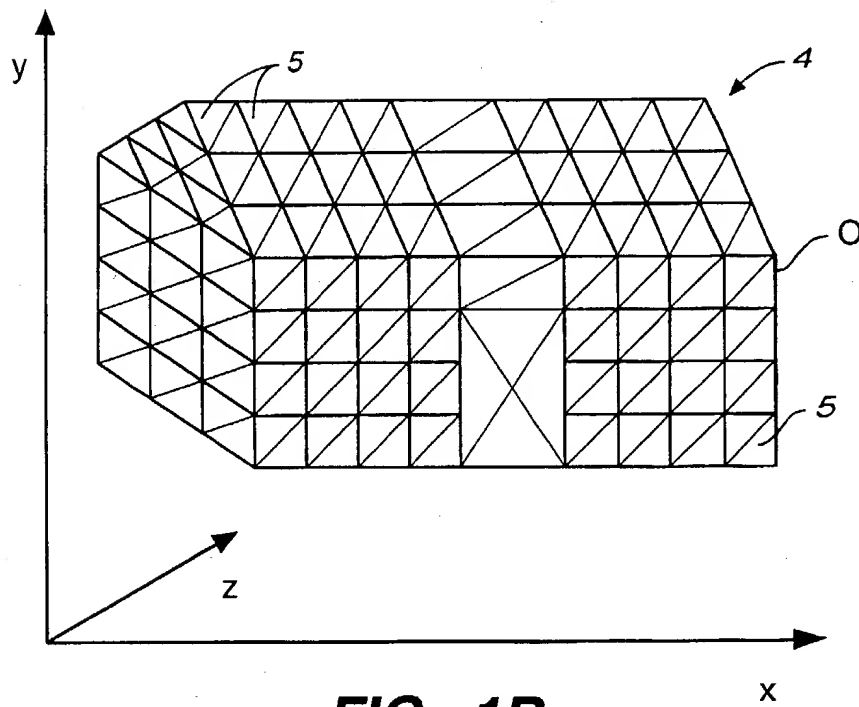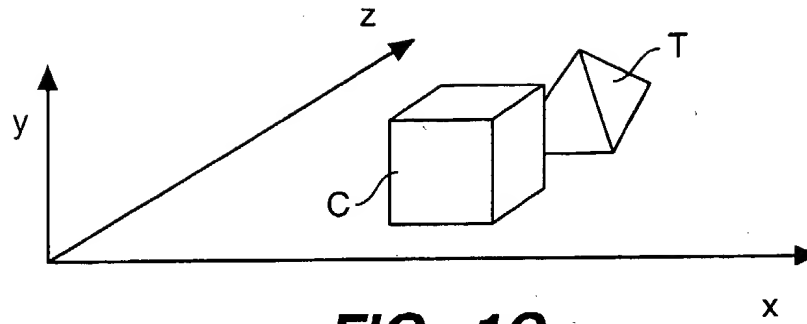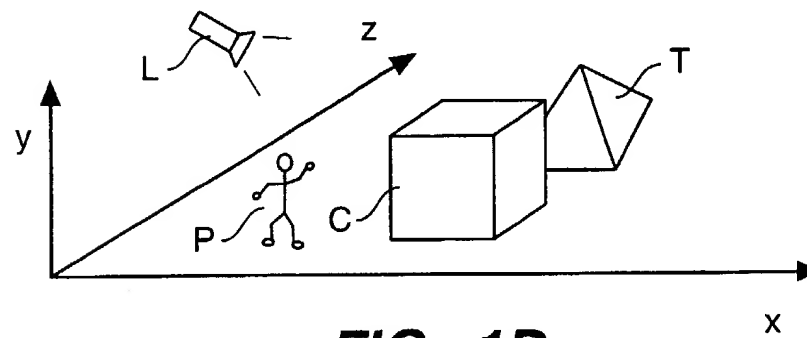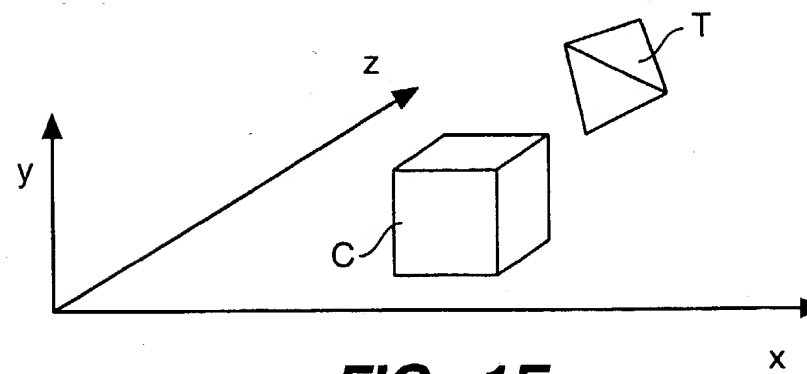
FIG._1A

FIG._1B

FIG._1C

FIG._1D

FIG._1E
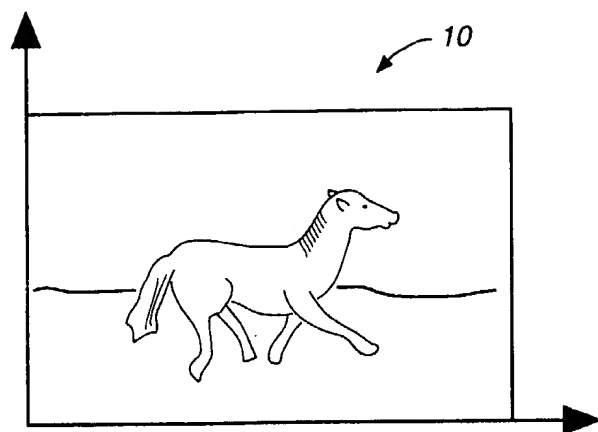
FIG._2A
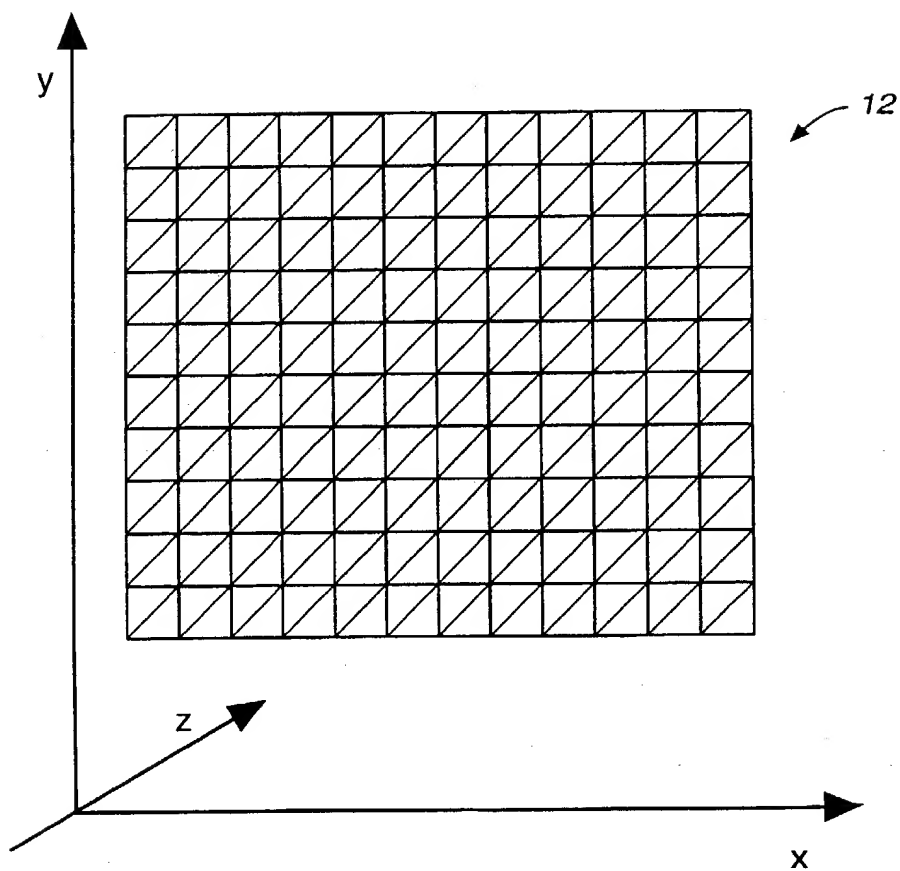
FIG._2B
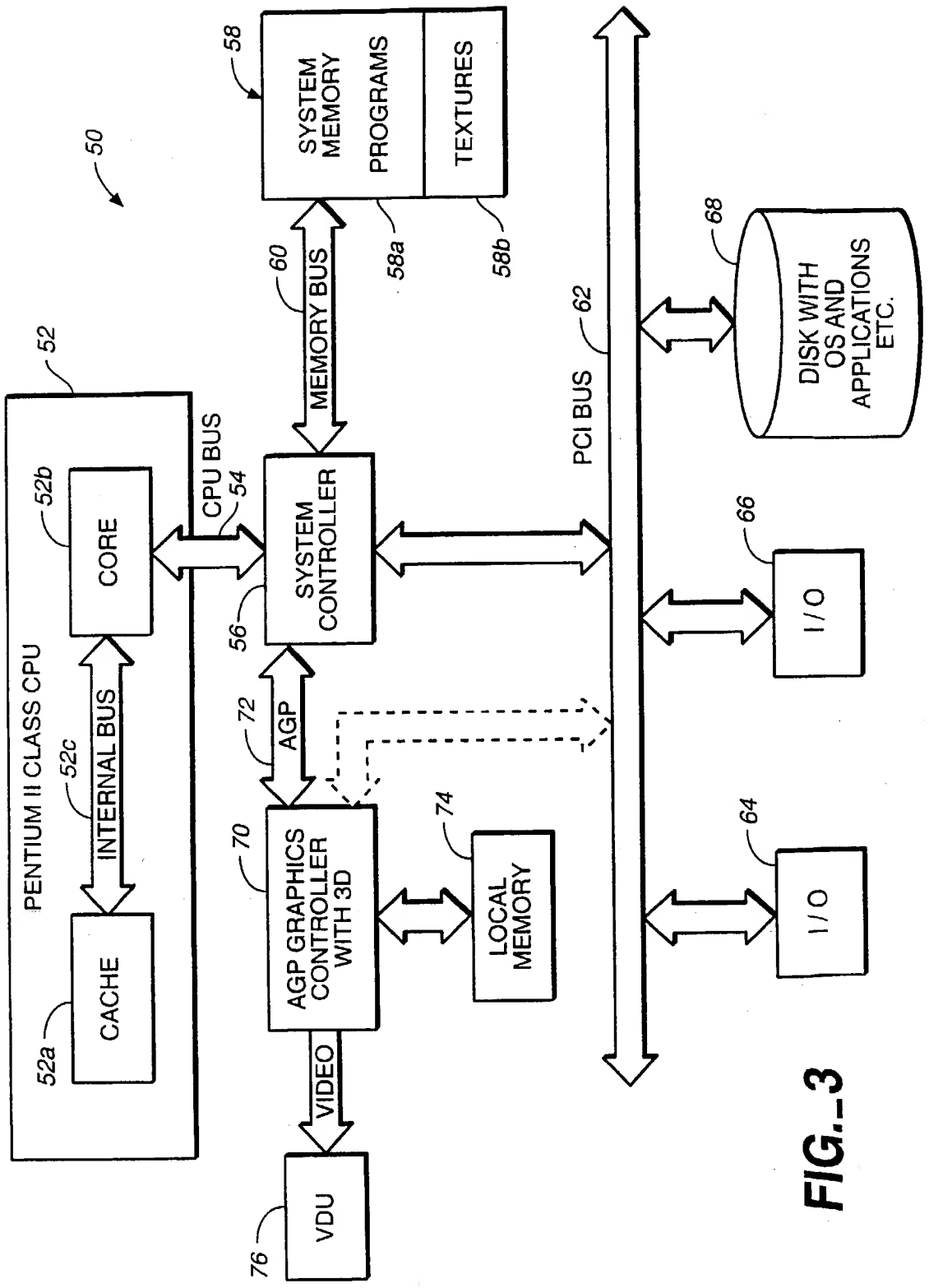
FIG._3

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06T 11/40

US CL :345/425

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/425, 423, 430, 435

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6,011,581 A (SWIFT et al) 04 January 2000, whole document	1-27
Y	US 5,369,735 A (THIER et al) 29 November 1994, whole document	1-27
Y	US 5,991,437 A (MIGDAL et al) 23 November 1999, whole document	1-27
Y	SEDERBERG, THOMAS W. & CHEN, FALAI. Implicitization using moving curves and surfaces Computer Graphics and Interactive Techniques Conference, Los Angeles, CA, August 1995. p 301-308	8

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 MAY 2000	**12 JUN 2000**
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MANO PADMANABHAN Telephone No. (703) 306-2903

Form PCT/ISA/210 (second sheet) (July 1998)*